IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Alexander, III et al.	§	Group Art Unit: 2193
	§	
Serial No.: 10/777,872	§	Examiner: Vu, Tuan A.
	§	
Filed: February 12, 2004	§	Confirmation No.: 6694
	§	
For: Method and Apparatus for Identifying Differences in Runs of a Computer Program Due to Code Changes	§	Attorney Docket No.: AUS920030820US1

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

FINAL PROPOSED AMENDMENTS

Sir:

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

<u>Amendments to the Claims:</u>

This listing of claims will replace all prior versions, and listings, of claims in the application:

<u>Listing of Claims:</u>

1.   (Currently Amended) A computer implemented method for identifying differences between an execution of a first built computer program and a second built computer program, comprising:

obtaining a first call tree data structure corresponding to first trace data of an execution of the first built computer program;

obtaining a second call tree data structure corresponding to second trace data of another execution of the second built computer program;

walking the second call tree data structure over the first call tree data structure to generate a third call tree data structure, wherein the third call tree data structure includes all nodes of both the first call tree data structure and the second call tree data structure, and wherein each node of the third call tree data structure includes a pass field having one of a first pass field value indicating that a first node was only present in the first call tree data structure, a second pass field value indicating that a second node was only present in the second call tree data structure, and a third pass field value indicating that a third node was present in both the first call tree data structure and the second call tree data structure, <u>wherein walking the second call tree data structure over the first call tree data structure further comprises:</u>

<u>for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the third call tree data structure with a pass field value computed by subtracting a base value of the node in the second call tree data structure from a base value of a corresponding node in the first call tree data structure;</u>

<u>for each node that exists in only the second call tree data structure, creating a node in the third call tree data structure with a pass field value having a negative base value corresponding to a base value of the node that exists in only the second call tree data structure;</u>

<u>setting the pass field of the node in the third call tree data structure to the third pass field value;</u> and

outputting the third call tree data structure, wherein the third call tree data structure identifies differences between the execution of the first built computer program and the another execution of the second built computer program.

2.   (Original) The method of claim 1, further comprising:

inputting the first trace data and the second trace data to an arcflow tool, wherein the arcflow tool generates the first call tree data structure and the second call tree data structure based on the first trace data and the second trace data.

3.   (Original) The method of claim 1, wherein the first call tree data structure and the second call tree data structure are xtree data structures.

4.   (Previously Presented) The method of claim 1, further comprising:

inserting the pass field in each node of the first call tree data structure; and

initializing the pass field in each node of the first call tree data structure to have the first pass field value.

5-7.   (Canceled).

8.   (Previously Presented)   The method of claim 6 1, further comprising:

setting   the pass field of the node in the third call tree data structure to the second pass field value.

9.   (Previously Presented) The method of claim 4, wherein nodes in the first call tree data structure and nodes in the second call tree data structure whose paths have not changed between builds are not present in the third call tree data structure.

10.   (Previously Presented) The method of claim 1, further comprising:

identifying improvements or regressions from the first built computer program to the second built computer program based on values associated with nodes in the third call tree data structure.

11.   (Currently Amended) A computer program product in a computer readable storage medium for identifying differences between the execution of a first built computer program and a second built computer program, comprising:

first instructions for obtaining a first call tree data structure corresponding to first trace data of an execution of the first built computer program;

second instructions for obtaining a second call tree data structure corresponding to second trace data of another execution of the second built computer program;

third instructions for walking the second call tree data structure over the first call tree data structure to generate a third call tree data structure, wherein the third call tree data structure includes all nodes of both the first call tree data structure and the second call tree data structure, and wherein each node of the third call tree data structure includes a pass field having one of a first pass field value indicating that a first node was only present in the first call tree data structure, a second pass field value indicating that a second node was only present in the second call tree data structure, and a third pass field value indicating that a third node was present in both the first call tree data structure and the second call tree data structure, wherein walking the second call tree data structure over the first call tree data structure further comprises:

instructions for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the third call tree data structure with a pass field value computed by subtracting a base value of the node in the second call tree data structure from a base value of a corresponding node in the first call tree data structure;

instructions for each node that exists in only the second call tree data structure, creating a node in the third call tree data structure with a pass field value having a negative base value corresponding to a base value of the node that exists in only the second call tree data structure;

instructions for setting the pass field of the node in the third call tree data structure to the third pass field value; and

fourth instructions for outputting the third call tree data structure, wherein the third call tree data structure identifies differences between the execution of the first built computer program and the another execution of the second built computer program.

12.   (Original) The computer program product of claim 11, further comprising:

fifth instructions for inputting the first trace data and the second trace data to an arcflow tool, wherein the arcflow tool generates the first call tree data structure and the second call tree data structure based on the first trace data and the second trace data.

13.   (Original) The computer program product of claim 11, wherein the first call tree data structure and the second call tree data structure are xtree data structures.

14.   (Previously Presented) The computer program product of claim 11, further comprising:

sixth instructions for inserting the pass field in each node of the first call tree data structure; and

seventh instructions for initializing the pass field in each node of the first call tree data structure to have the first pass field value.

15-17. (Canceled).

18.   (Previously Presented) The computer program product of claim ~~16~~ 11, further comprising:

instructions for setting the pass field of the node in the third call tree data structure to the second pass field value.

19.   (Previously Presented) The computer program product of claim 11, further comprising:

eighth instructions for identifying improvements or regressions from the first built computer program to the second built computer program based on values associated with nodes in the third call tree data structure.

20.   (Currently Amended) An apparatus for identifying differences between the execution of a first built computer program and a second built computer program, comprising:

a computer for executing the following means, further comprising:

means for obtaining a first call tree data structure corresponding to first trace data of an execution of the first build of the computer program;

means for obtaining a second call tree data structure corresponding to second trace data of another execution of the second build of the computer program;

means for walking the second call tree data structure over the first call tree data structure to generate a third call tree data structure, wherein the third call tree data structure includes all nodes of both the first call tree data structure and the second call tree data structure, and wherein each node of the third call tree data structure includes a pass field having one of a first pass field value indicating that a first node was only present in the first call tree data structure, a second pass field value indicating that a second node was only present in the second call tree data structure, and a third pass field value indicating that a third node was present in both the first call tree data structure and the second call tree data structure, wherein walking the second call tree data structure over the first call tree data structure further comprises:

means for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the third call tree data structure with a pass field value computed by subtracting a base value of the node in the second call tree data structure from a base value of a corresponding node in the first call tree data structure;

means for each node that exists in only the second call tree data structure, creating a node in the third call tree data structure with a pass field value having a negative base value corresponding to a base value of the node that exists in only the second call tree data structure;

means for setting the pass field of the node in the third call tree data structure to the third pass field value; and

means for outputting the third call tree data structure, wherein the third call tree data structure identifies differences between the execution of the first built computer program and the another execution of the second built computer program.

DATE: September 30, 2008

Respectfully submitted,

/Houda El-Jarrah/

Houda El-Jarrah
Reg. No. 60,006
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicant